UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/810,208	03/26/2004	Yuen Fai Wong	FOUND-0091 (034103-000036	8639

49680 7590 07/16/2007

FOUNDRY-THELEN REID BROWN RAYSMAN & STEINER LLP
P.O. BOX 640640
SAN JOSE, CA 95164-0640

EXAMINER	
PATEL, CHANDRAHAS B	

ART UNIT	PAPER NUMBER
2616	

MAIL DATE	DELIVERY MODE
07/16/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/810,208	WONG ET AL.	
	Examiner	Art Unit	
	Chandrahas Patel	2616	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 November 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-70* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-70* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 July 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Specification

1. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1, 2, 4-11, 14, 18, 20-28, 31-34, 36, 37, 39-44, 46-48, 50-52, 54, 55, 57-62, 64-70 are rejected under 35 U.S.C. 102(e) as being anticipated by Maher, III et al. (USPN 6,654,373).

Regarding claim 1, Maher, III teaches a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the circuit, the first processors, and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the circuit comprising: a plurality of ingress data ports, each ingress data port adapted to receive an input data stream from a corresponding first processor, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; an aggregation module coupled to the plurality of ingress data ports, the aggregation module adapted to analyze and combine the plurality of input data steams into one aggregated data stream in response to the priority factors **[Fig. 2, 140, Col.**

6, lines 15-25]; a memory coupled to the aggregation module, the memory adapted to store analyzed data packets **[Fig. 2, 112]**; and an output data port coupled to the aggregation module, the output data port adapted to output the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claims 2, 21, 40, 58, Maher, III teaches each of the first processors and the second processor transmits and receives a data stream through a logical interface providing logical interconnection between a Media Access Control sublayer (MAC) and a Physical layer (PHY) **[Fig. 2, 102, 120, Abstract]**.

Regarding claim 4, Maher, III teaches the second processor is a data packet processor **[Fig. 4, 418]**.

Regarding claims 5, 22, 39, 57, Maher, III teaches memory is an external buffer memory **[Fig. 2, 112]**.

Regarding claims 6, 14, Maher, III teaches an egress data input port adapted to receive a data stream from the second processor, the data stream formed of egress data packets **[Fig. 4, 414 outputs the data to 404 through single port]**; a plurality of egress data output ports, each adapted to output an output data stream to a corresponding one of the first processors **[Fig. 4, multiple outputs shown going out of 404 to 402]**; and a forwarding module coupled between the egress data input port and the egress data output port, the forwarding module adapted to forward an egress data packet in the data stream from the second processor to one of the egress data output port in response to destination information associated with the egress data packet **[Fig. 4, 404, Col. 11, lines 14-17]**.

Regarding claim 7, Maher, III teaches ingress data ports include a first data port for receiving a first input data stream and a second data port for receiving a second input data stream **[Fig. 2, 102, Col. 6, lines 5-10]**; and aggregation module includes: a first packet analyzer coupled to the first data port, adapted to classify each of the ingress data packets in the first data stream into one of predetermined priority classes based on the priority factors **[Fig. 2, 110, Col. 7, lines 41-44]**; a second packet analyzer coupled to the second data port, adapted to classify each of the ingress data packets in the second data stream into one of predetermined priority classes based on the priority factors **[Fig. 2, 110, Col. 7, lines 41-44]**; a queue module having a plurality of priority queues each provided for the corresponding priority class, adapted to store a packet descriptor of each of the analyzed data packets classified to the corresponding priority class, the packet descriptor containing a reference to a memory location of its analyzed data packet in memory, and a selection logic implementing a queue scheme, adapted to arbitrate and select a packet descriptor from among the priority queues **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; a first write interface coupled to first packet analyzer, adapted to write the analyzed data packets into the memory at the memory location indicated by the corresponding packet descriptor **[Col. 7, lines 34-36]**; a second write interface coupled to second packet analyzer, adapted to write the analyzed data packets into memory at the memory location indicated by the corresponding packet descriptor **[Col. 7, lines 34-36]**; a common read interface coupled to queue selection logic, adapted to read a data packet corresponding to the selected packet descriptor from memory **[Col. 8, lines 7-14]**; and an output module to send the data packets read from memory to output data port as the aggregated data stream **[Fig. 2, 120]**.

Regarding claims 8, 24, 33, 51, Maher, III teaches packet analyzer includes a data decoder coupled to ingress data port to decode a header of each ingress data packet to extract the priority factors **[Fig. 2, 104, Col. 6, lines 15-25]**.

Regarding claims 9, 25, Maher, III teaches a read buffer is coupled to common read interface **[Fig. 2, 136]**.

Regarding claims 10, 26, 37, 55, Maher, III teaches a data encoder is coupled to read buffer, that encodes the data packets into an interface format corresponding to the first interface before sending from the output port **[Col. 8, lines 26-28]**.

Regarding claims 11, 28, Maher, III teaches a write buffer coupled between first packet analyzer and first write interface **[Fig. 2, 118]**.

Regarding claim 18, Maher, III teaches a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the circuit, the first processors, and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the circuit comprising: a plurality of ingress data ports, each ingress data port adapted to receive an input data stream from a corresponding first processor, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; an aggregation module coupled to the plurality of ingress data ports, the aggregation module adapted to analyze and combine the plurality of input data steams into one aggregated data stream in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; a memory coupled to the aggregation module, the memory adapted

to store analyzed data packets **[Fig. 2, 112]**; and an output data port coupled to the aggregation module, the output data port adapted to output the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claim 20, Maher, III teaches a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the circuit, the first processors, and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the circuit comprising: an ingress data port adapted to receive the input data stream from the first processor via a first data link having a first bandwidth, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; an aggregation module coupled to ingress data port, the aggregation module adapted to analyze and selectively recombine the ingress data packets in response to the priority factors so as to generate an aggregated data stream for a second data link having a second bandwidth smaller than the first bandwidth **[Col. 7, lines 47-52]**; a memory coupled to the aggregation module, the memory adapted to store analyzed data packets **[Fig. 2, 112]**; and an output data port coupled to the aggregation module, the output data port adapted to output the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claim 23, Maher, III teaches aggregation module comprising: a packet analyzer adapted to classify each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 110, Col. 7, lines 41-44]**; a queue module comprising a plurality of priority queues each provided for the corresponding priority class, adapted to store a packet descriptor of each of the analyzed data packets classified to the

corresponding priority class, the packet descriptor containing a reference to a memory location of its analyzed data packet in the memory, and a selection logic implementing a queue scheme, adapted to arbitrate and select a packet descriptor from among the priority queues **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; a read interface coupled to the queue module, adapted to read a data packet corresponding to the selected packet descriptor from the memory **[Col. 8, lines 7-14]**; and an output module to send the data packets read from the memory to the output data port as the aggregated data stream **[Fig. 2, 120]**.

Regarding claim 27, Maher, III teaches a write interface coupled to the packet analyzer, adapted to write the analyzed data packets into the memory at the memory location indicated by the corresponding packet descriptor **[Col. 7, lines 34-46]**.

Regarding claim 31, Maher, III teaches a circuit for generating a plurality of output data streams for first processors from an aggregated input data stream from a second processor, the circuit, the first processors, and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the circuit comprising: an egress data input port adapted to receive the aggregated input data stream from the second processor, the aggregated data stream formed of egress data packets **[Fig. 4, 414 outputs the data to 404 through single port]**; a plurality of egress data output ports, each adapted to output an output data stream to a corresponding one of the first processors **[Fig. 4, multiple outputs shown going out of 404 to 402]**; and a forwarding module coupled between the egress data input port and the egress data output port, the forwarding module adapted to forward an egress data packet in the data stream from the second processor to one of the egress data output port in response to destination information associated with the egress data packet **[Fig. 4, 404, Col. 11, lines 14-17]**.

Regarding claim 32, Maher, III, teaches a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; storing an analyzed data packet in a memory **[Fig. 2, 112]**; generating a packet descriptor for the analyzed ingress data packet, the packet descriptor containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; placing the packet descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; arbitrating and selecting a packet descriptor from among the priority queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**; reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Regarding claim 34, 52, Maher, III teaches buffering the analyzed data packet in a write buffer before storing in the memory **[Fig. 2, 118]**.

Regarding claim 36, 54, Maher, III teaches buffering the data packet read from the memory in a read buffer **[Fig. 2, 118]**.

Regarding claim 41, 59, Maher, III teaches analyzing and classifying, generating, and storing are performed separately for each data stream **[Col. 7, lines 38-40, Fig. 2, 110]**.

Regarding claim 42, 60, Maher, III teaches packet descriptors from each stream of a same priority class are placed in the same priority queue for that priority class **[Col. 7, lines 41-53]**.

Regarding claim 43, 61, Maher, III teaches arbitrating and selecting, reading, and sending are performed as a single data channel **[Fig. 2, Path 126]**.

Regarding claim 44, 62, Maher, III teaches protocol filtering to determine if the ingress data packet is a certain protocol packet **[Col. 6, lines 22-25]**.

Regarding claim 46, Maher, III teaches a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; generating an aggregated data stream by analyzing and combining the plurality of input data steams into one aggregated data stream in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and outputting the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claim 47, Maher, III teaches a method for aggregating data packets received from a first processor for a second processor, the first processor being capable of outputting data

with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving the input data stream from the first processor via a first data link having the first bandwidth, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; generating an aggregated data stream by analyzing and selectively recombining the ingress data packets in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and outputting the aggregated data stream to the second processor via a second data link having the second bandwidth **[Col. 7, lines 47-52]**.

Regarding claim 48, Maher, III teaches a method for aggregating data packets received from a first processor for a second processor, the first processor being capable of outputting data with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from the first processors, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; storing an analyzed data packet in a memory **[Fig. 2, 112]**; generating a packet descriptor for the analyzed ingress data

packet, the packet descriptor containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; placing the packet descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; arbitrating and selecting a packet descriptor from among the priority queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**; reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Regarding claim 50, Maher, III, teaches an apparatus for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the apparatus comprising: means for receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]** including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; means for analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; means for storing an analyzed data packet in a memory **[Fig. 2, 112]**; means for generating a packet descriptor for the analyzed ingress data packet, the packet descriptor containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; means for placing the packet descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; means for arbitrating and selecting a packet descriptor from among the priority

queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**; means for reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and means for sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Regarding claim 64, Maher, III teaches an apparatus for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the apparatus comprising: means for receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; means for generating an aggregated data stream by analyzing and combining the plurality of input data steams into one aggregated data stream in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and means for outputting the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claim 65, Maher, III teaches an apparatus for aggregating data packets received from a first processor for a second processor, the first processor being capable of outputting data with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2, 100]**, the apparatus comprising: means for receiving the input data stream from the first processor via a first data link having the first

bandwidth, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; means for generating an aggregated data stream by analyzing and selectively recombining the ingress data packets in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and means for outputting the aggregated data stream to the second processor via a second data link having the second bandwidth **[Col. 7, lines 47-52]**.

Regarding claim 66, Maher, III teaches an apparatus for aggregating data packets received from a first processor for a second processor, the first processor being capable of outputting data with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2, 100]**, the apparatus comprising: means for receiving an input data stream from the first processors, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; means for analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; means for storing an analyzed data packet in a memory **[Fig. 2, 112]**; means for generating a packet descriptor for the analyzed ingress data packet, the packet descriptor containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; means for placing the packet

descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; means for arbitrating and selecting a packet descriptor from among the priority queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**; means for reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and means for sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Regarding claim 67, Maher, III a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine **[Col. 11, lines 33-41]** to perform a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; storing an analyzed data packet in a memory **[Fig. 2, 112]**; generating a packet descriptor for the analyzed ingress data packet, the packet descriptor containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; placing the packet descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; arbitrating and selecting a packet descriptor from among the priority queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**;

reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Regarding claim 68, Maher, III teaches a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine **[Col. 11, lines 33-41]** to perform a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from each of the first processors, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; generating an aggregated data stream by analyzing and combining the plurality of input data steams into one aggregated data stream in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and outputting the aggregated data stream to the second processor **[Fig. 2, 120]**.

Regarding claim 69, Maher, III teaches a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine **[Col. 11, lines 33-41]** to perform a method for aggregating data packets received from a first processor for a second processor, the first processor being capable of outputting data with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2,**

100], the method comprising: receiving the input data stream from the first processor via a first data link having the first bandwidth, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; generating an aggregated data stream by analyzing and selectively recombining the ingress data packets in response to the priority factors, the priority factors including an indication of whether the ingress packet contains protocol data or not **[Fig. 2, 140, Col. 6, lines 15-25]**; and outputting the aggregated data stream to the second processor via a second data link having the second bandwidth **[Col. 7, lines 47-52]**.

Regarding claim 70, Maher, III teaches a program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine **[Col. 11, lines 33-41]** to perform a method for aggregating data packets received from a first processor for a second processor, the first processor is capable of outputting data with a first bandwidth greater than a second bandwidth **[Col. 7, lines 47-52]** by which the second processor is capable of receiving the data, the first and second processors being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: receiving an input data stream from the first processors, the input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; analyzing and classifying each of the ingress data packets into one of predetermined priority classes based on the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; storing an analyzed data packet in a memory **[Fig. 2, 112]**; generating a packet descriptor for the analyzed ingress data packet, the packet descriptor

containing a reference to a memory location of its analyzed data packet stored in the memory **[Fig. 2, 116, packets are stored in 132, Col. 7, lines 34-44]**; placing the packet descriptor in a priority queue corresponding to the priority class of the data packet **[Col. 7, lines 41-44]**; arbitrating and selecting a packet descriptor from among the priority queues using selection logic implementing a queue scheme **[Col. 7, lines 18-28]**; reading a data packet corresponding to the selected packet descriptor from the memory **[Col. 7, lines 36-40]**; and sending the data packets read from the memory to the second processor as an aggregated data stream **[Col. 8, lines 7-14]**.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Maher, III et al. (USPN 6,654,373) in view of Kadambi et al. (USPN 6,335,935).

Regarding claim 3, Maher, III teaches the circuit as discussed in rejection of claim 1.

However, Maher, III does not teach the first processors are Layer-2 switching processors.

Kadambi teaches the first processors are Layer-2 switching processors **[Fig. 2, 20, Col. 6, lines 4-10]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have Layer-2 switching processors as first processor since it has to handle complex data for properly process a significant number of different types of packets **[Col. 6, lines 13-18]**.

6. Claims 12, 13, 15, 29, 30, 35, 38, 53, 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maher, III et al. (USPN 6,654,373) in view of Fedorkow et al. (USPN 7,230,917).

Regarding claims 12, 13, 15, 29, 30, 35, 38, 53, 56, Maher, III teaches a circuit, a method, an apparatus as discussed in rejection of claim 11, 7, 14, 28, 23, 34, 32, 52, 50 respectively.

However, Maher, III does not teach a flow control module is adapted to assert a flow control if an amount of data stored in buffer exceeds a threshold.

Fedorkow teaches a flow control module is adapted to assert a flow control if an amount of data stored in buffer exceeds a threshold **[Abstract]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to assert a flow control signal if buffer exceeds a threshold so that it could be determined if more packets can be stored in the buffer or not **[Abstract]**.

7. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Maher, III et al. (USPN 6,654,373) in view of Fedorkow et al. (USPN 7,230,917) as applied to claim 15 above, and further in view of Manaka et al. (USPN 6,421,352).

Regarding claim 16, the references teach a circuit as discussed in rejection of claim 15.

However, the references do not teach inserting a pause control packet when queue exceeds a threshold.

Manaka teaches inserting a pause control packet when queue exceeds a threshold **[Col. 2, lines 25-28]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to insert a pause control packet when congestion occurs so that only one of the LAN connections is interrupted instead of all **[Col. 4, lines 62-66]**.

8. Claims 17, 49 is rejected under 35 U.S.C. 103(a) as being unpatentable over Maher, III et al. (USPN 6,654,373) in view of Abbas et al. (USPN 6,810,046).

Regarding claim 17, Maher, III teaches a circuit for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the circuit, the first processors, and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the circuit comprising: a plurality of ingress data ports, each ingress data port adapted to receive an input data stream from a corresponding first processor, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; an aggregation module coupled to the plurality of ingress data ports, the aggregation module adapted to analyze and combine the plurality of input data steams into one aggregated data stream in response to the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**; a memory coupled to the aggregation module, the memory adapted to store analyzed data packets **[Fig. 2, 112]**; and an output data port coupled to the aggregation module, the output data port adapted to output the aggregated data stream to the second processor **[Fig. 2, 120]**.

However, Maher, III does not teach a field programmable logic device (FPLD) implements the aggregation module.

Abbas teaches a FPLD implements the aggregation module **[Fig. 3, 540]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have a FPLD implement the aggregation module so that configuration of the aggregation module can be changed after it has been fabricated and can be programmed to match the need of system.

Regarding claim 49, Maher, III teaches a method for aggregating a plurality of input data streams from first processors into one data stream for a second processor, the first processors and the second processor being provided on an electronic circuit substrate **[Fig. 2, 100]**, the method comprising: providing a *device* coupled between the first processors and second processors **[Fig. 2, 140]**; providing an ingress data interface between each of the first processors and the *device*, each ingress data interface being adapted to couple an input data stream from a corresponding first processor to the *device*, each input data stream formed of ingress data packets **[Fig. 2, 102, Col. 6, lines 5-14]**, each ingress data packet including priority factors coded therein **[Col. 6, lines 22-25, type or protocol can be a priority factor as discussed in applicant's specification]**; providing a memory coupled to the *device*, the memory adapted to store analyzed data packets **[Fig. 2, 112]**; providing an output data interface between the *device* and the second processor, the output data interface being adapted to couple the aggregated data stream to the second processor **[Fig. 2, 120]**; and programming the *device* such that the *device* analyzes and combines the plurality of input data steams into one aggregated data stream in response to the priority factors **[Fig. 2, 140, Col. 6, lines 15-25]**.

However, Maher, III does not teach that the device is a field programmable logic device (FPLD).

Abbas teaches that FPLD can act as such device **[Fig. 3, 540]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have a FPLD implement the device so that configuration of the aggregation module can be changed after it has been fabricated and can be programmed to match the need of system.

9. Claims 19, 45, 63 is rejected under 35 U.S.C. 103(a) as being unpatentable over Maher, III et al. (USPN 6,654,373) in view of Mackiewich et al. (USPN 7,212,536).

Regarding claim 19, Maher, III teaches priority factors include per-port priority **[Col 3, lines 34-37]**.

However, Maher, III does not teach priority factors include VLAN priority.

Mackiewich teaches priority factors include VLAN priority **[Abstract]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have VLAN priority so that different treatment can be given to connections with different priority **[Col. 1, lines 7-11]**.

Regarding claims 45, 63, Maher, III teaches priority factors comprise protocol filter priority **[Col. 1, lines 6-8]**, per-port priority **[Col 3, lines 34-37]**.

However, Maher, III does not teach priority factors include VLAN priority.

Mackiewich teaches priority factors include VLAN priority **[Abstract]**.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to have VLAN priority so that different treatment can be given to connections with different priority **[Col. 1, lines 7-11]**.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chandrahas Patel whose telephone number is 571-270-1211. The examiner can normally be reached on Monday through Thursday 7:30 to 17:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ricky Ngo can be reached on 571-272-3139. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

CBP

RICKY Q. NGO
SUPERVISORY PATENT EXAMINER